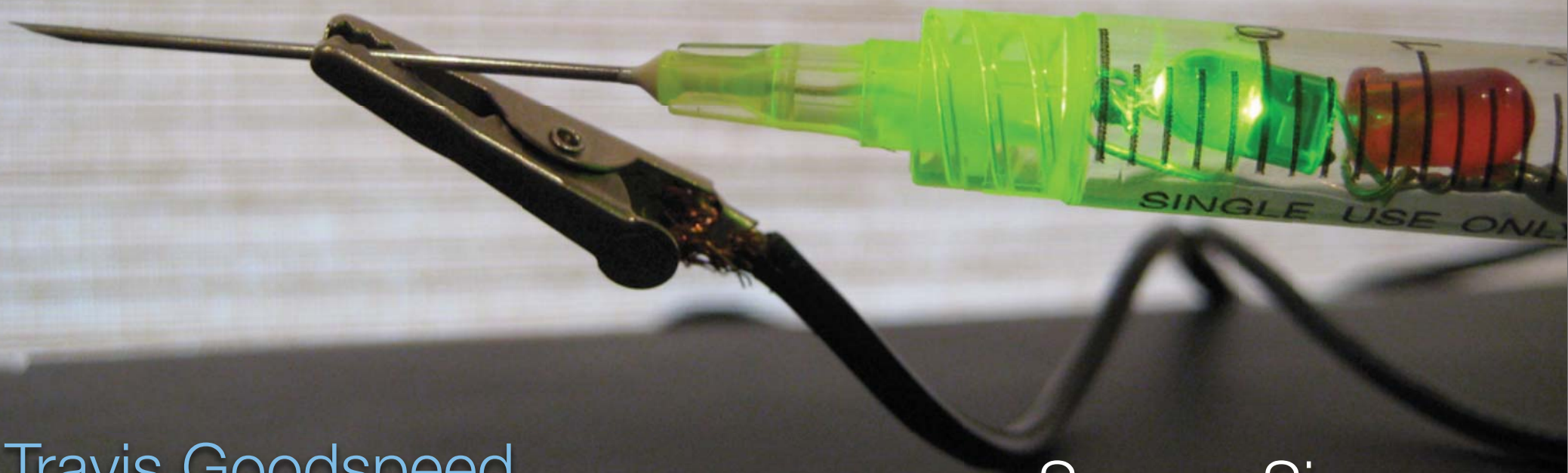


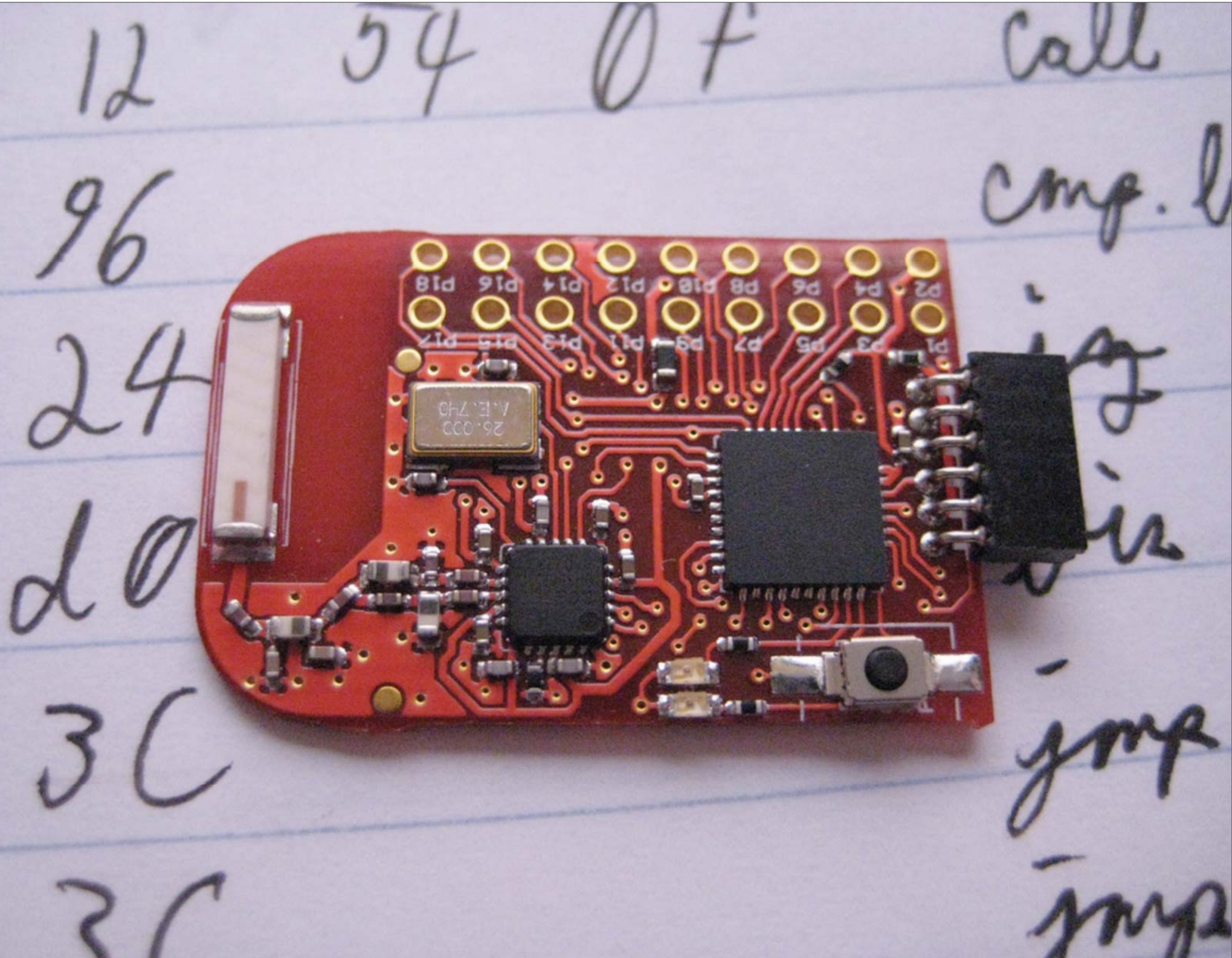
Hacking Sub-GHz Radio



Travis Goodspeed

[<travis@radiantmachines.com>](mailto:travis@radiantmachines.com)

Syscan Singapore
July 2010



Brief

- ✧ Lots of LPAN Radio Standards
 - ✧ IEEE802.15.4, ANT, SimpliciTI, etc at 2.4GHz
 - ✧ 15.4, SimpliciTI, etc at 800, 900 MHz
 - ✧ Dash7 and others at 433MHz
- ✧ How to sniff all of them, inexpensively?

Brief

- ✦ Lots of LPAN standards.
- ✦ How to sniff all of them, inexpensively?
 - ✦ Reuse of prebuilt modules.
 - ✦ Custom modules where needed.
 - ✦ Chipsets do the work.

Hardware Platforms

- ✦ USB Attached
 - ✦ Telos B, Zolertia Z1
 - ✦ CC1111, CC2531 Modules
- ✦ Standalone
 - ✦ Girttech IM ME
 - ✦ Next Hope Badge

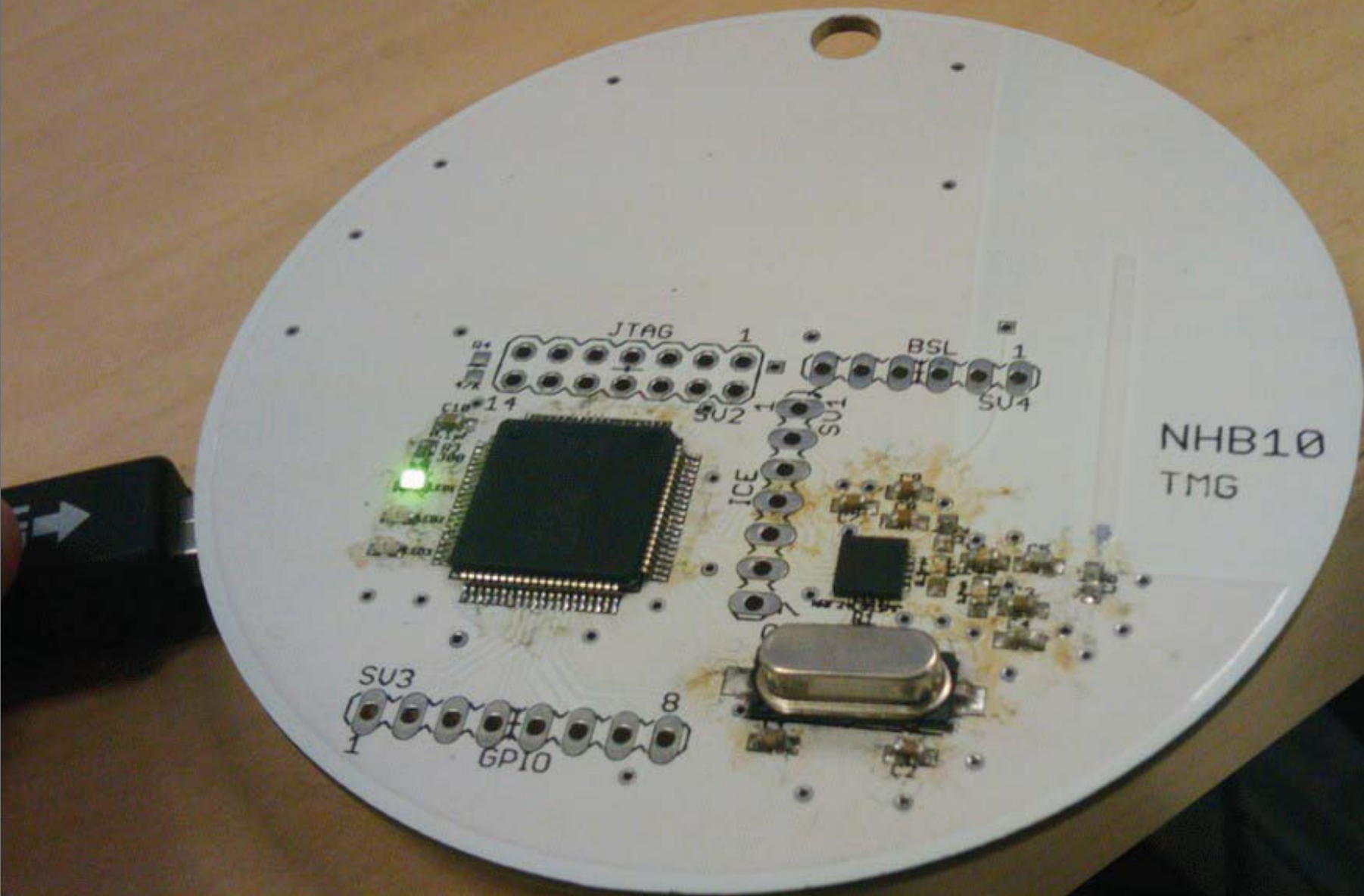


Telos B

- ✦ Development Platform
- ✦ Widely cloned, manufactured for use by academia.
- ✦ Many commercial designs are a quick copy of this.



Next Hope Badge



Software Applications

- ✦ TinyOS, Contiki
- ✦ OpenBeacon
- ✦ Girttech IM ME

OpenBeacon, HOPE Badge

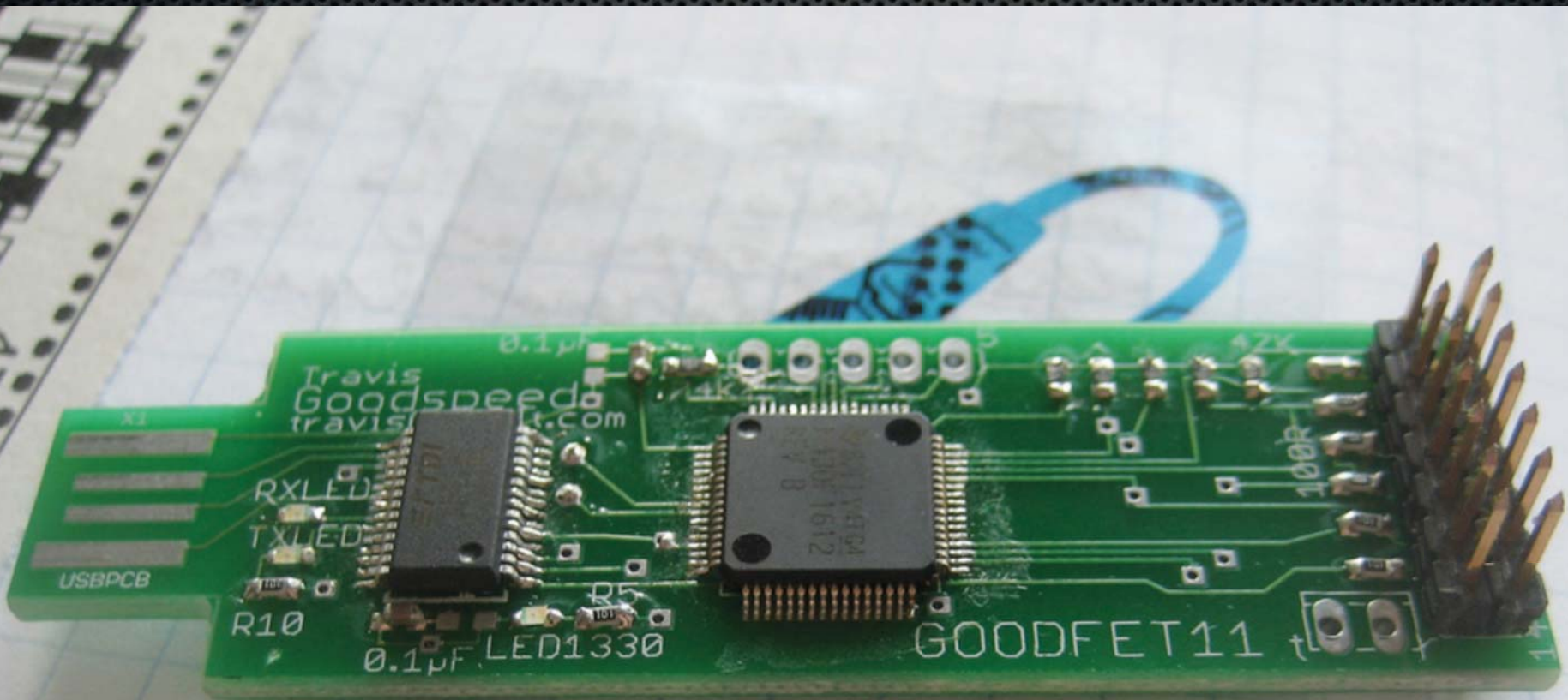
- ✧ Beacons on 2.481GHz using Nordic NRF24L01+
 - ✧ Easy to tap, intercept.
- ✧ XXTEA Encryption, Shared Symmetric Key
 - ✧ Held in MSP430, not NRF24L01+.
 - ✧ Single, shared key for the entire network.
- ✧ Reflashable with new firmware.

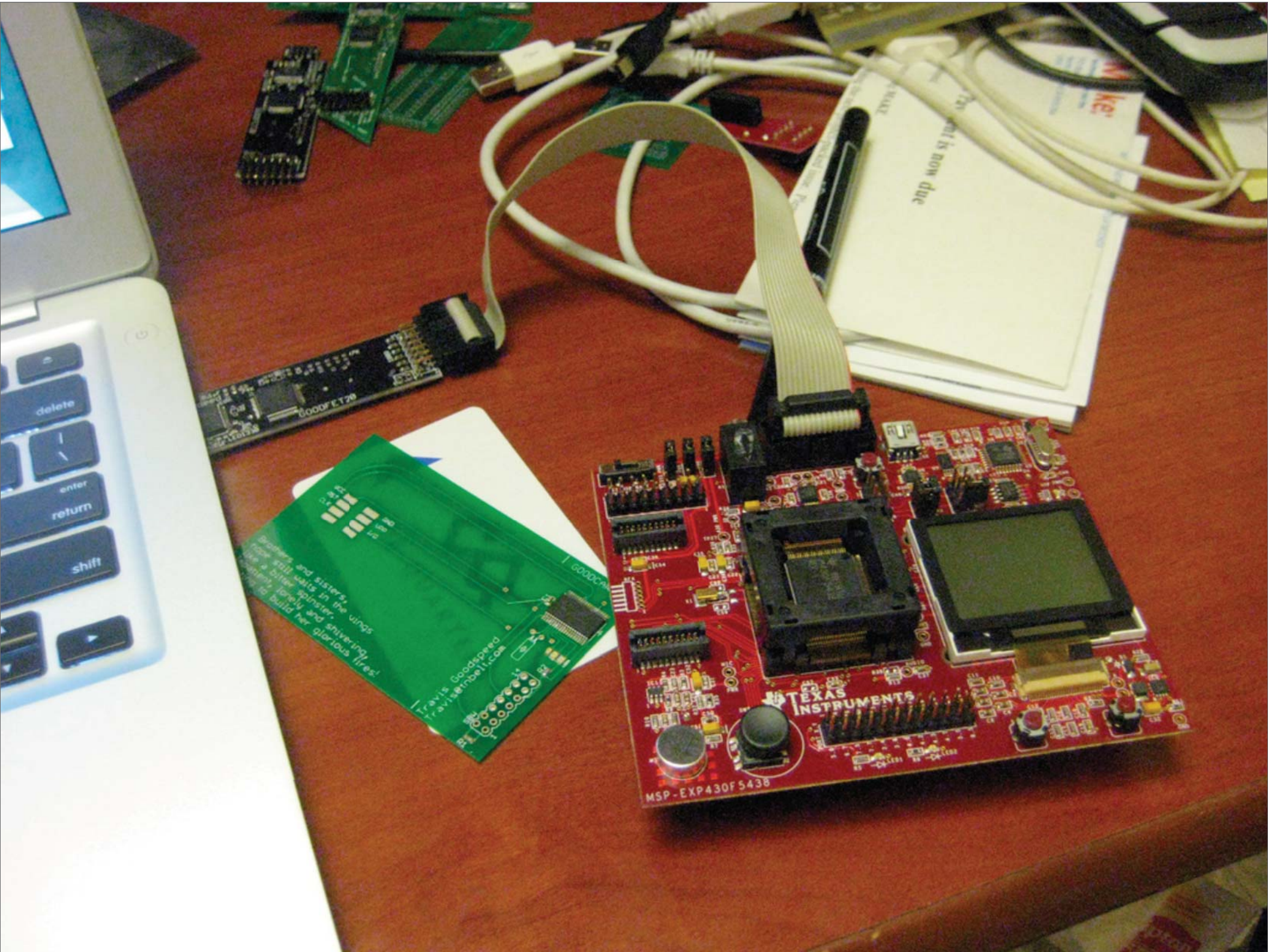
Girltech IM ME

- ✧ Single CC1110
 - ✧ Both a radio and a microcontroller.
 - ✧ Debugger exposes RAM, but not Flash.
- ✧ Reflash with new firmware.
 - ✧ Mike Ossmann's \$15 Spectrum Analyzer

GoodFET

- ✦ Universal hardware debugger.
- ✦ Client in Python for rapid access to target hardware.





Friday, August 13, 2010

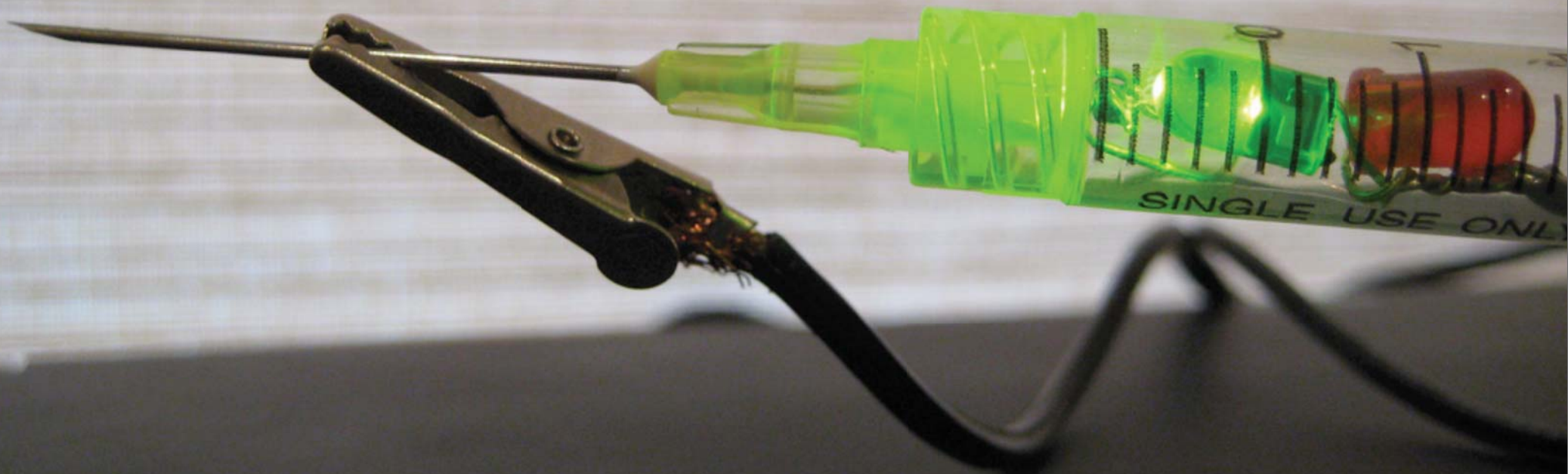
GoodFET Interface

- ✦ Abstracts chipset protocols to register access.
 - ✦ SPI, I2C, JTAG, and 8051
 - ✦ “reg[5]=0xCAFE”
 - ✦ “ram[3A0]=0xDEADBEEFDEADBEEF”
- ✦ Works on prebuilt devices through debugging port!

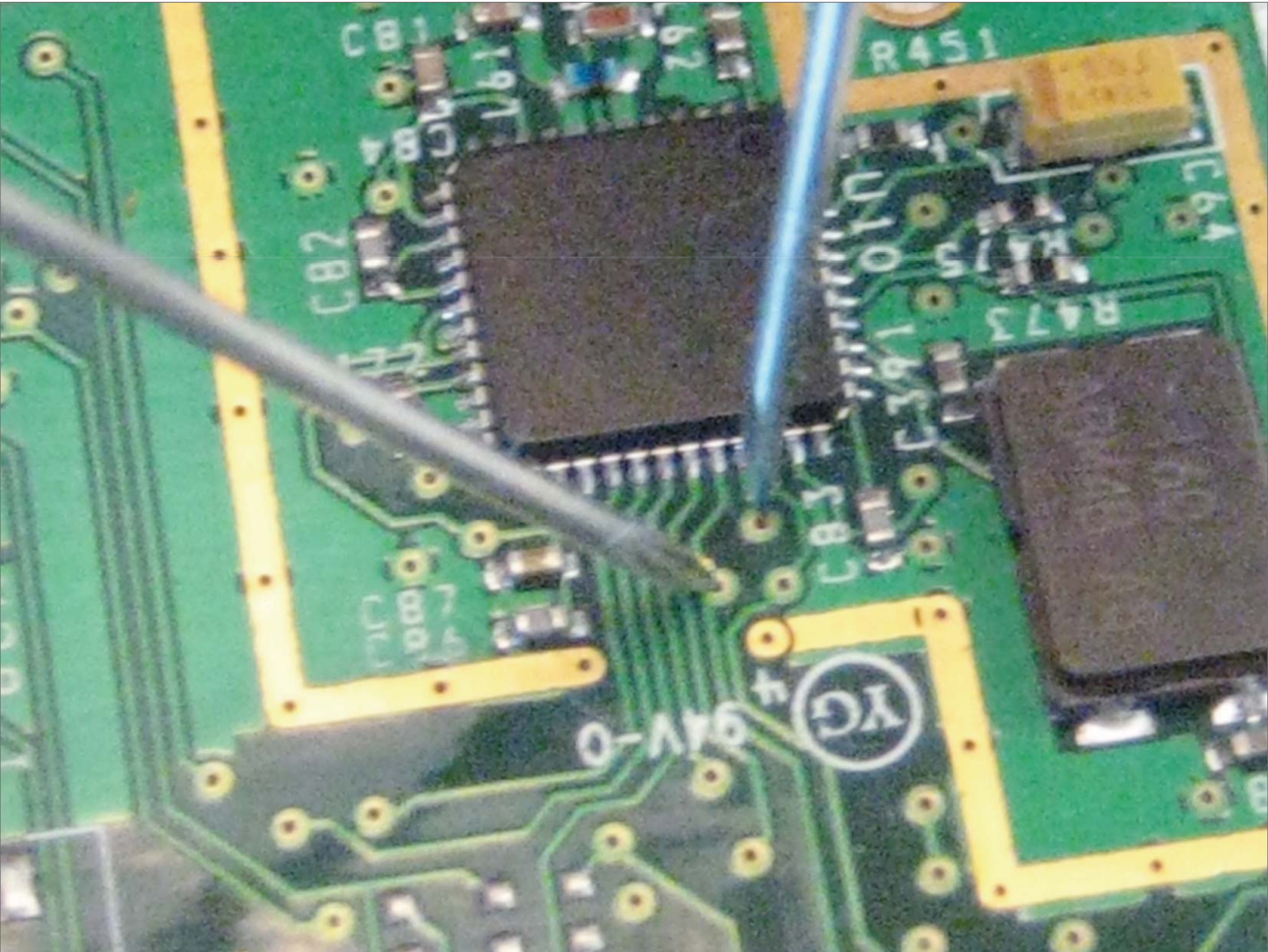
Attacks

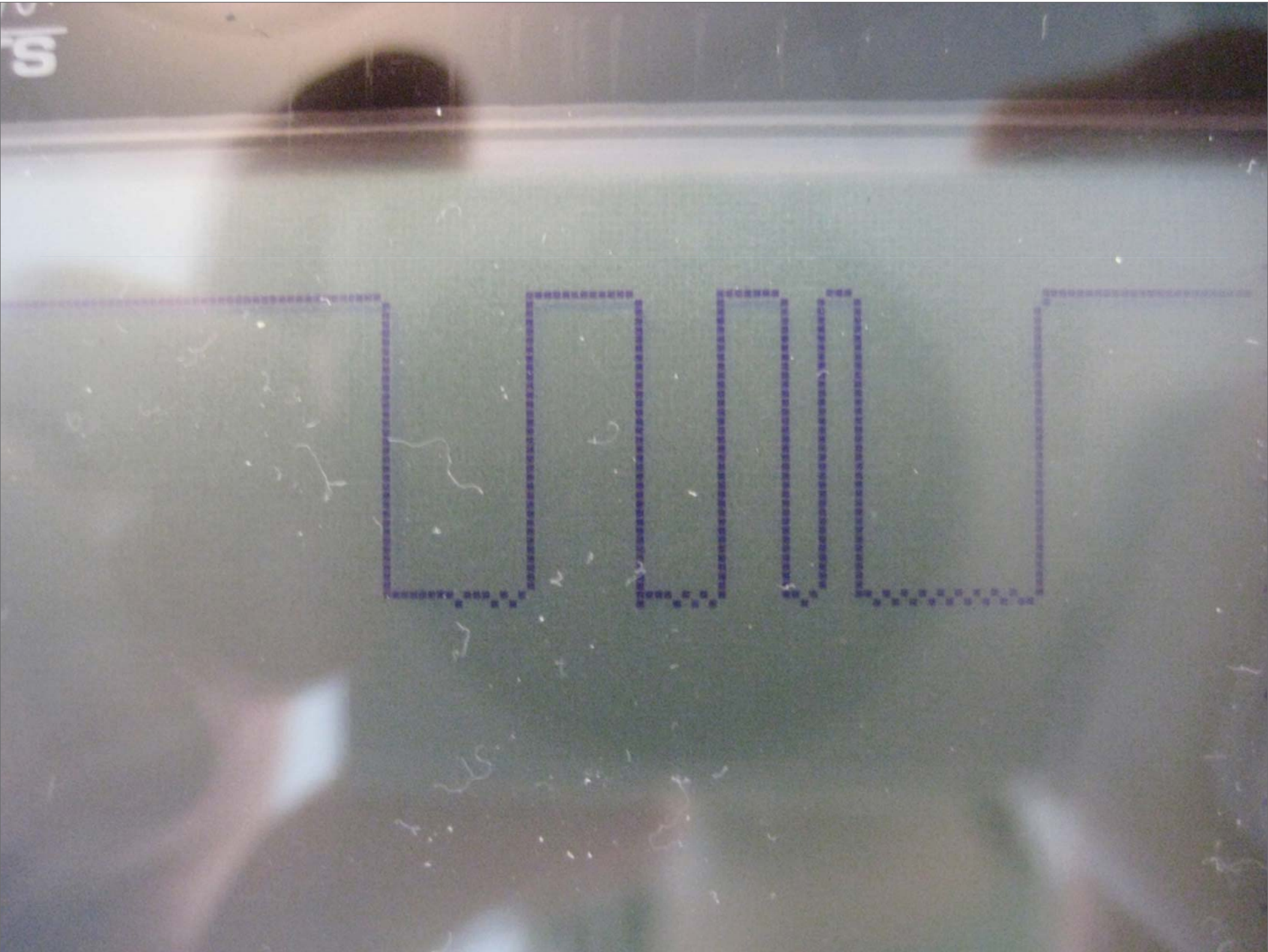
- ✦ Radio Bus Sniffing
- ✦ Radio Bus Replacement
- ✦ Rapid Driver Development

Radio Bus Sniffing









SPI Sniffing

Total Phase - Beagle Data Center - /opt/beagle-gui/source09_demo.bgl

File Beagle Help

Connection disconnected Settings Capture 43 KB Scroll Hide Filters

All USB I2C SPI MDIO

Index	m:s.ms.us	Dur	Prot	Len	Err	Data
10	2:18.286.005	5.43 s	SPI	358	TMP4	0000 0000 0000 0000 0000 0000 0000 0000
11	2:25.366.651	1.33 s	SPI	82	TMP3	0046 0046 0046 0046 0046 0046 0046 0046
12	2:29.273.215	1.74 s	SPI	114	TMP6	0046 0046 0046 0046 0046 0046 0046 0046
13	2:34.156.570	1.31 s	SPI	79	TMP2	0088 0031 0022 0000 00FF 00FF 0001 0000
14	2:39.772.105	2.29 s	SPI	147	TMP2	003E 000E 0041 0088 0048 0022 0000 0000
15	2:44.655.310	2.52 s	SPI	163	TMP4	003E 000E 0041 0088 005C 0022 0000 0000
16	2:50.270.995	35.4 s	SPI	2357	TMP2	003E 000E 0041 0088 0073 0022 0000 0000

General Data Filters

Minimum index:

Maximum index:

Apply Filters Reset Filters

Find: Prev Next

☒ Ascii ☐ Hex ☐ Meta

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
230	00FF	00F0	0010	0000	0060	00D9	0068	0028	0004	00F8	0039	0006	0022	0058	0088	0003
240	00FF	00FC	0004	0000	0018	0036	005E	000A	0001	003E	000E	0041	0088	0097	0022	0000
250	00FF	00FF	0001	0000	0006	000D	0098	0082	0080	004F	0083	0090	0062	0026	0008	0080
260	003F	00FF	00C0	0040	0001	0083	0066	0060	00A0	0013	00E0	00E4	0018	0089	0092	0020
270	000F	00FF	00F0	0010	0000	0060	00D9	00A8	0028	0004	00F8	0039	0006	0022	0068	0088
280	0003	00FF	00FC	0004	0000	0018	0036	006E	000A	0001	003E	000E	0041	0088	009B	0022
290	0000	00FE	00FF	0001	0000	0006	000D	009C	0082	0080	004F	0083	0090	0062	0027	0008

Data Timing

Ready. Beagle disconnected.

Radio Bus Replacement

- ✦ A GoodFET can usurp control of a radio chip by
 - ✦ 1) Holding the master of an SPI bus in reset.
 - ✦ 2) Erasing and reflashing a microcontroller.

Hope Badge Hack

- ✦ 1) Connect the badge by the USB port.
- ✦ 2) Run 'goodfet.bsl --fromweb'
 - ✦ Erases OpenBeacon firmware, installs GoodFET.
- ✦ 3) Run 'goodfet.nrf regs >state.txt'
 - ✦ Records state for future use.
- ✦ 4) Swap MAC addresses, run 'goodfet.nrf sniff'.

Hope Badge Hack

```
air-2% goodfet.nrf info
Encoding GFSK
Freq          2481 MHz
Rate          2000 kbps
PacketLen 16 bytes
SMAC 0x0102030201
TMAC 0x424541434f
air-2%
```


Hope Badge Hack

```
air-2% goodfet.nrf sniffob
Listening as 0102030201 on 2481 MHz
10 17 ff aa 00 00 00 02 be ef de ad 00 00 1f b3
10 17 ff ff 00 00 00 03 be ef de ad 00 00 3f 34
10 17 bf ff 00 00 00 0f be ef de ad 00 00 b0 1f
10 17 bf ff 00 00 00 13 be ef de ad 00 00 d5 0f
10 17 b9 ff 00 00 00 23 be ef de ad 00 00 87 bc
10 17 a9 ff 00 00 00 2b be ef de ad 00 00 23 41
10 17 bf 55 00 00 00 2d be ef de ad 00 00 9f e7
10 17 bf aa 00 00 00 2e be ef de ad 00 00 fe 6e
10 17 bf ff 00 00 00 2f be ef de ad 00 00 de e9
10 17 bf 55 00 00 00 31 be ef de ad 00 00 fa f7
10 17 bf aa 00 00 00 32 be ef de ad 00 00 9b 7e
10 17 bf ff 00 00 00 33 be ef de ad 00 00 bb f9
10 17 bf 00 00 00 00 34 be ef de ad 00 00 1b b6
```


Rapid Driver Development

- ✦ There are a lot of radio chips out there.
 - ✦ Most of them are available as SPI.
 - ✦ Even minor models have different registers.
- ✦ XML can help us!

SmartRF 7

- ✦ Register Configuration Tool
 - ✦ Every Chipcon Radio
- ✦ Definitions in XML
 - ✦ Easy to parse.
 - ✦ Easy to apply.



SmartRF Chips

- ✦ SPI Bus
 - ✦ CC1100, CC2420
- ✦ Chipcon 8051 Debug Protocol
 - ✦ CC1110, CC2430
- ✦ MSP430 JTAG Debug Protocol
 - ✦ CC430

SmartRF Configuration

Typical Settings

Data rate: 1.2 kBaud, Dev.: 5.2 kHz, Mod.: GFSK, RX BW: 58 kHz, Optimized for sensitivity
Data rate: 1.2 kBaud, Dev.: 5.2 kHz, Mod.: GFSK, RX BW: 58 kHz, Optimized for current consumption
Data rate: 2.4 kBaud, Dev.: 5.2 kHz, Mod.: GFSK, RX BW: 58 kHz, Optimized for sensitivity
Data rate: 2.4 kBaud, Dev.: 5.2 kHz, Mod.: GFSK, RX BW: 58 kHz, Optimized for current consumption
Data rate: 38.4 kBaud, Dev.: 20 kHz, Mod.: GFSK, RX BW: 100 kHz, Optimized for sensitivity
Data rate: 38.4 kBaud, Dev.: 20 kHz, Mod.: GFSK, RX BW: 100 kHz, Optimized for current consumption
Data rate: 250 kBaud, Dev.: 127 kHz, Mod.: GFSK, RX BW: 540 kHz, Optimized for sensitivity

RF Parameters

Base Frequency 868.299866 MHz	Channel Number 0	Channel Spacing 199.951172 kHz	Carrier Frequency 868.299866 MHz
Xtal Frequency 26.000000 MHz	Data Rate 1.19948 kBaud	RX Filter BW 58.035714 kHz	☐ Manchester Enable
Modulation Format GFSK	Deviation 5.157471 kHz	TX Power 0 dBm	☐ PA Ramping

Register Export

Export Format

Select Registers

Comment Delimiters

Header

RN rn RC Rd rd AH ah AD VH vh VD <<

:

=

TAB

//

BS

CLEAR

Registers

`sfr @RN@= @<<@0x@AH@; @<<@// @Rd@`

Footer

Templates

C51 SFR definitions

Open

C51 SFR definitions
html
Packet sniffer settings
RF settings SoC
RF settings struct typedef
RF settings
SimpliciTI settings

Save

Delete

Code export

Preview

Help

```
sfr PSCTR11= 0xDF07; // Frequency Synthesizer
sfr PSCTR10= 0xDF08; // Frequency Synthesizer
sfr FRRQ2= 0xDF09; // Frequency Control Word
sfr FRRQ1= 0xDF0A; // Frequency Control Word
sfr FRRQ0= 0xDF0B; // Frequency Control Word
sfr MDMCFG4= 0xDF0C; // Modem configuration
sfr MDMCFG3= 0xDF0D; // Modem Configuration
sfr MDMCFG2= 0xDF0E; // Modem Configuration
sfr MDMCFG1= 0xDF0F; // Modem Configuration
sfr MDMCFG0= 0xDF10; // Modem Configuration
sfr CHANNR= 0xDF06; // Channel Number
sfr DEVTATN= 0xDF11; // Modem Deviation Settlin
sfr FRRND1= 0xDF1A; // Front End RX Configur
sfr FRRND0= 0xDF1B; // Front End TX Configur
sfr MCSM0= 0xDF14; // Main Radio Control Sta
sfr FOCCFG= 0xDF15; // Frequency Offset Compe
sfr BSCFG= 0xDF16; // Bit Synchronization Cc
sfr AGCCTR12= 0xDF17; // AGC Control
sfr AGCCTR11= 0xDF18; // AGC Control
sfr AGCCTR10= 0xDF19; // AGC Control
sfr PSCA13= 0xDF1C; // Frequency Synthesizer
sfr PSCA12= 0xDF1D; // Frequency Synthesizer
```

Export to File

☐ Make Device Specific

Copy to Clipboard

XML Definitions

```
<!--CC1100-->
<Device>
  <DeviceName>CC1100</DeviceName>
  <Brief> Sub-1GHz</Brief>
  <DeviceType>Transceiver</DeviceType>
</Device>
<!--CC1100E-->
<Device>
  <DeviceName>CC1100E</DeviceName>
  <Brief> Sub-1GHz.</Brief>
  <DeviceType>Transceiver</DeviceType>
</Device>
<!--CC1101-->
<Device>
  <DeviceName>CC1101</DeviceName>
  <Brief> Sub-1GHz</Brief>
  <DeviceType>Transceiver</DeviceType>
</Device>
```


XML Definitions

```
<!--RSSI-->
<Register>
  <Name>RSSI</Name>
  <Address>0xDF3A</Address>
  <Description>Received Signal Strength Indication </Description>
  <Bitfield>
    <Name>RSSI</Name>
    <Start>7</Start>
    <Stop>0</Stop>
    <Access>R</Access>
    <Reset>0x80</Reset>
    <Description><![CDATA[Received signal strength indicator]]></De
  </Bitfield>
  <RegReset>80</RegReset>
</Register>
```


SmartRF7/GoodFET

- ✦ The GoodFET client can use SmartRF register definitions.
- ✦ The GoodFET adapter supports all of the register transfer protocols.

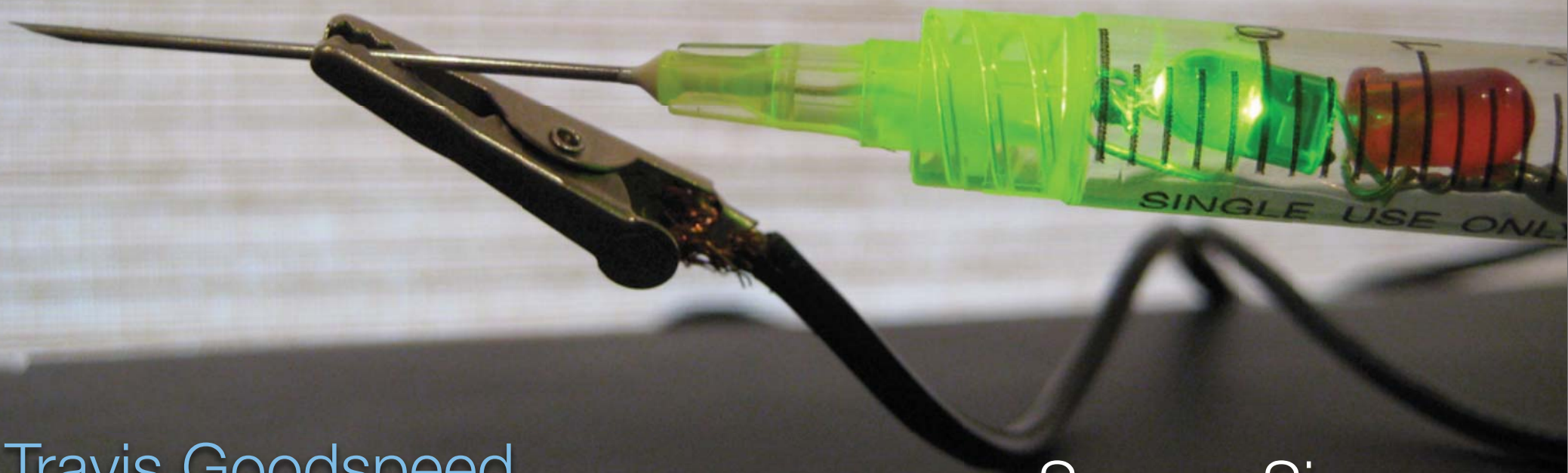
XML Definitions

```
<!--RSSI-->
<Register>
  <Name>RSSI</Name>
  <Address>0xDF3A</Address>
  <Description>Received Signal Strength Indication </Description>
  <Bitfield>
    <Name>RSSI</Name>
    <Start>7</Start>
    <Stop>0</Stop>
    <Access>R</Access>
    <Reset>0x80</Reset>
    <Description><![CDATA[Received signal strength indicator]]></De
  </Bitfield>
  <RegReset>80</RegReset>
</Register>
```


XML GoodFET Integration

```
air-2% goodfet.cc radioinfo | head -n 20
// CC1110 RadioState
SYNC1      =0xd3; /* Sync Word, High Byte
                /*      [7:0]                                SYNC[15:8]
SYNC0      =0x91; /* Sync Word, Low Byte
                /*      [7:0]                                SYNC[7:0]
PKTLEN     =0xff; /* Packet Length
                /*      [7:0]                                PACKET_LENGTH
PKTCTRL1   =0x04; /* Packet Automation Control
                /*      [7:5]                                PQT[2:0]
                /*      [4:3]                                unused
                /*      [2:2]                                APPEND_STATUS
                /*      [1:0]                                ADR_CHK[1:0]
PKTCTRL0   =0x45; /* Packet Automation Control
                /*      [7:7]                                unused
                /*      [6:6]                                WHITE_DATA
                /*      [5:4]                                PKT_FORMAT[1:0]
                /*      [3:3]                                unused
                /*      [2:2]                                CRC_EN
                /*      [1:0]                                LENGTH_CONFIG[1:0]
ADDR       =0x00; /* Device Address
```

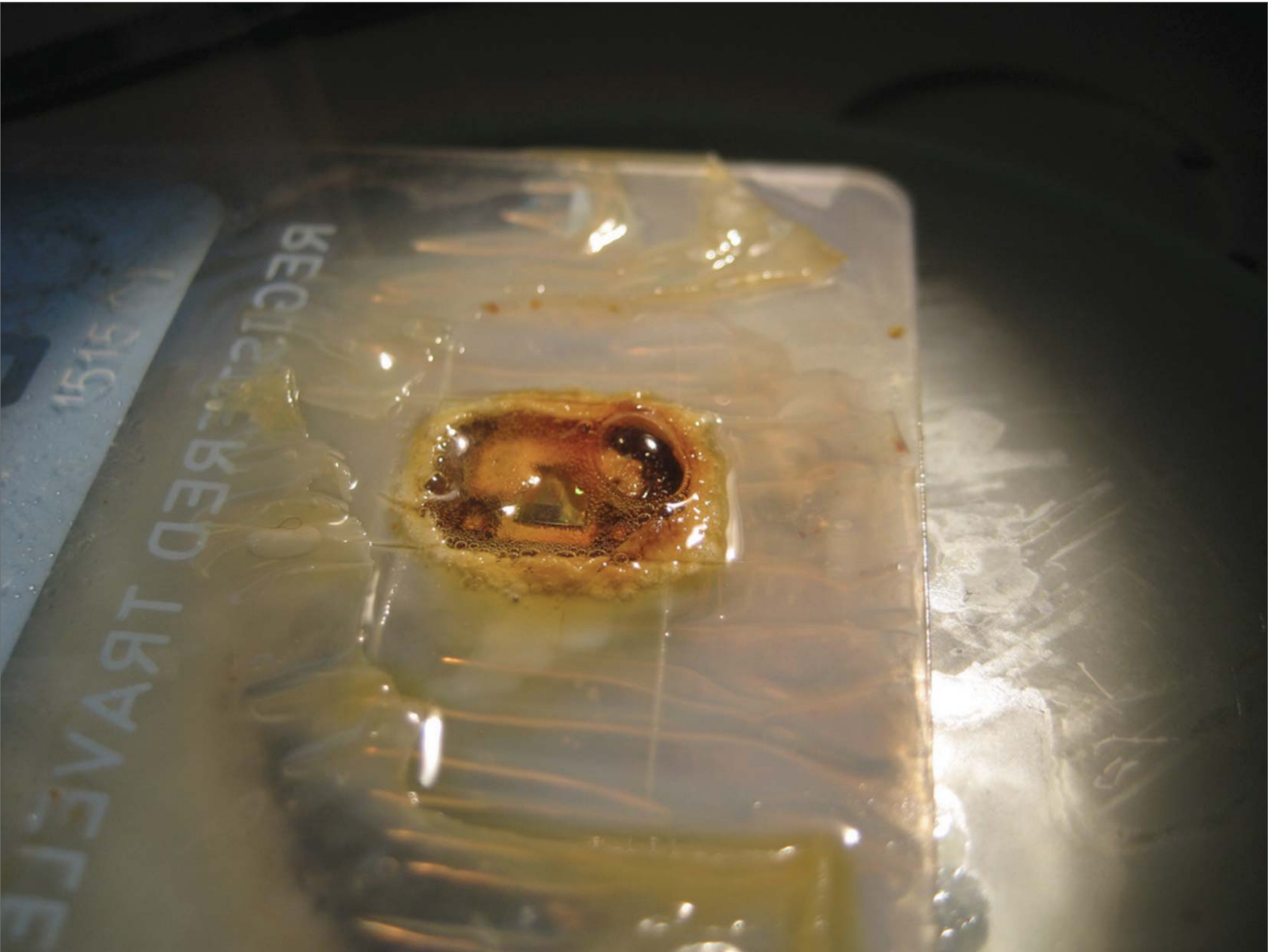

Hacking Sub-GHz Radio



Travis Goodspeed

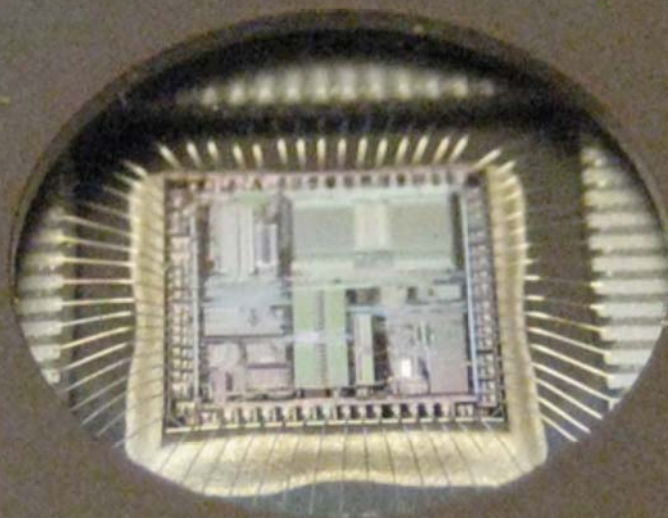
<travis@radiantmachines.com>

Syscan Singapore
July 2010





84AHL2T



PMS430E325FZ



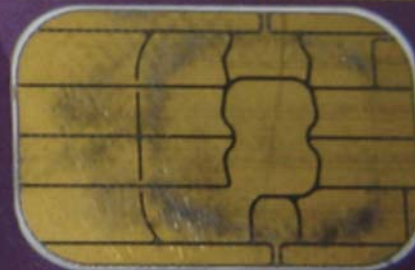
Le passe Navigo Découverte n'est valable qu'associé avec la carte nominative transport portant votre nom, prénom et photo, et le même numéro que le passe. Lors d'un contrôle, le passe et la carte nominative doivent être présentés ensemble, un justificatif d'identité peut être demandé.

135 283

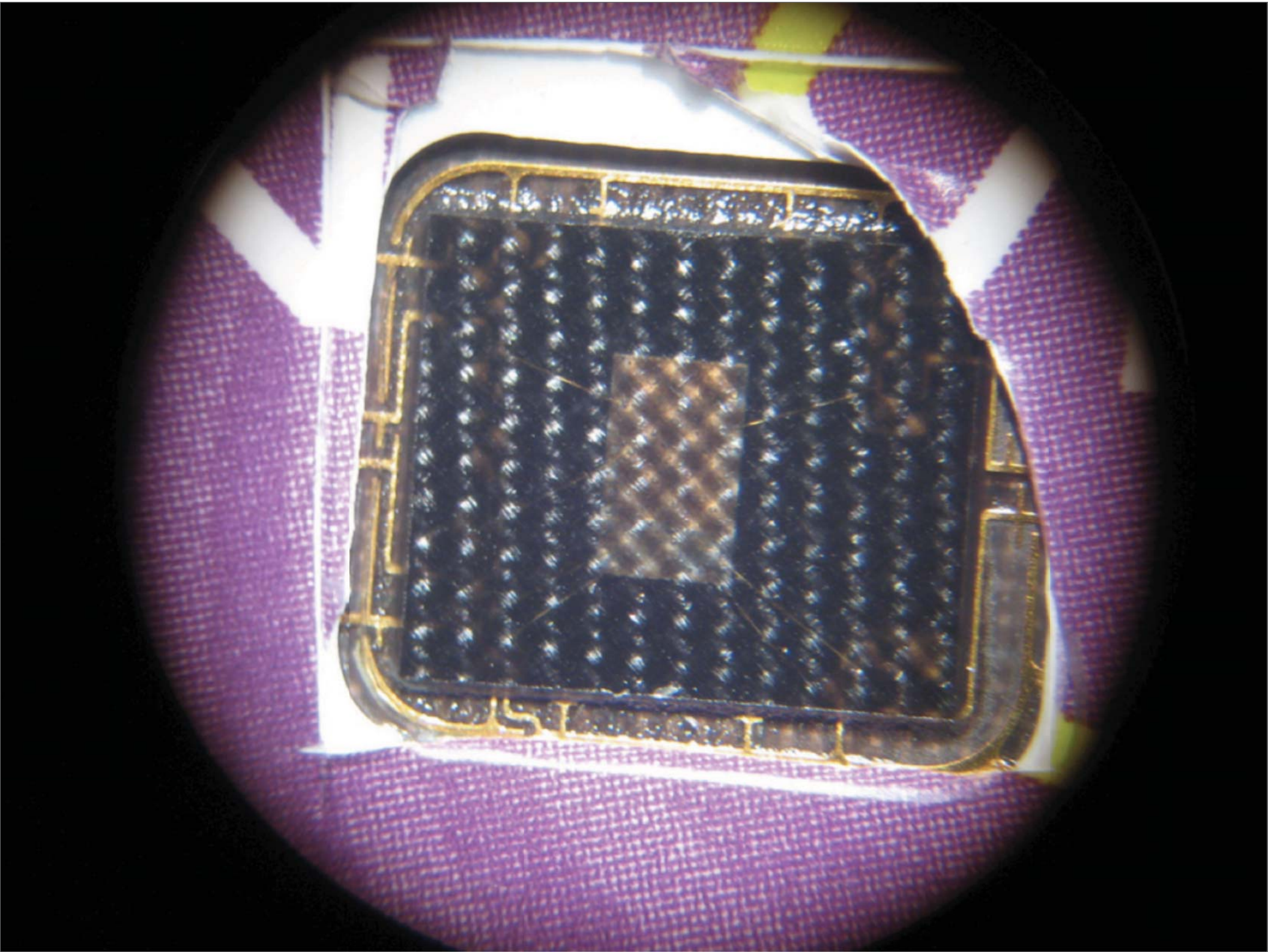
NAVI

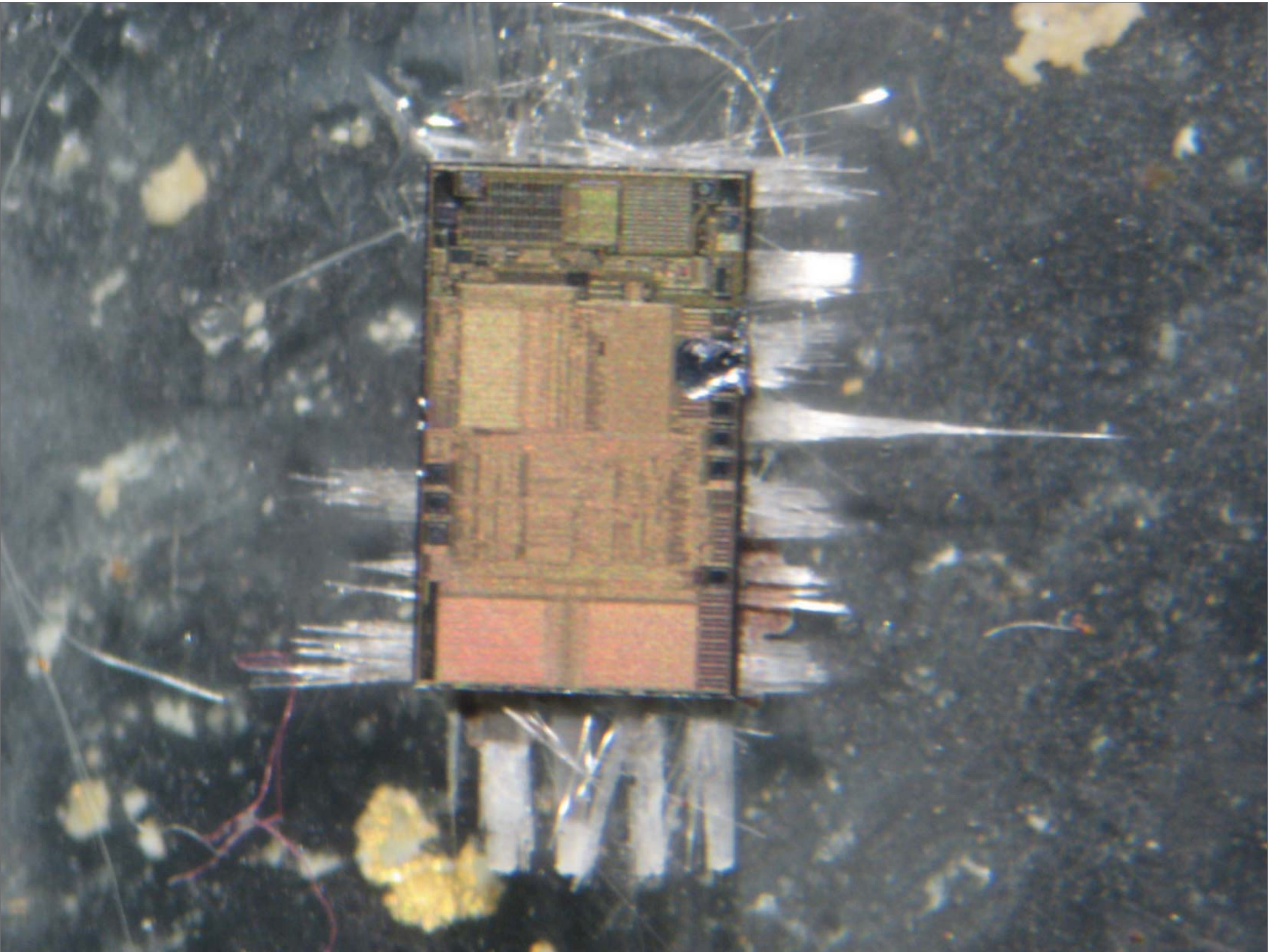


DÉCOUVERTE

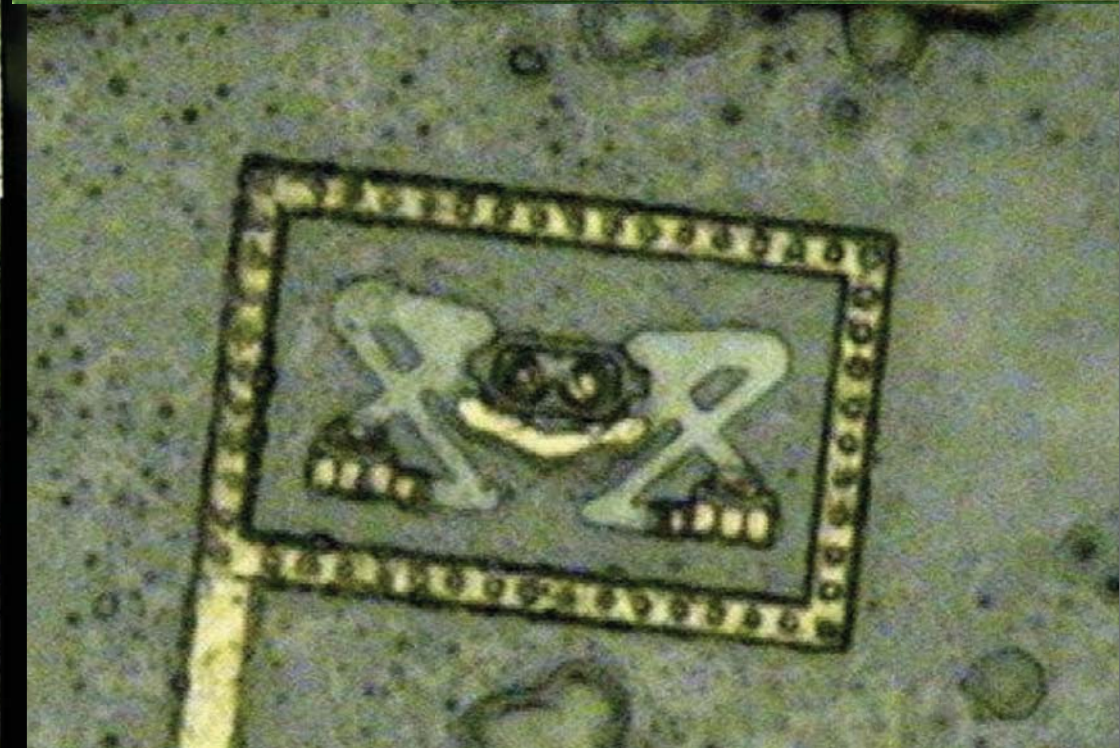
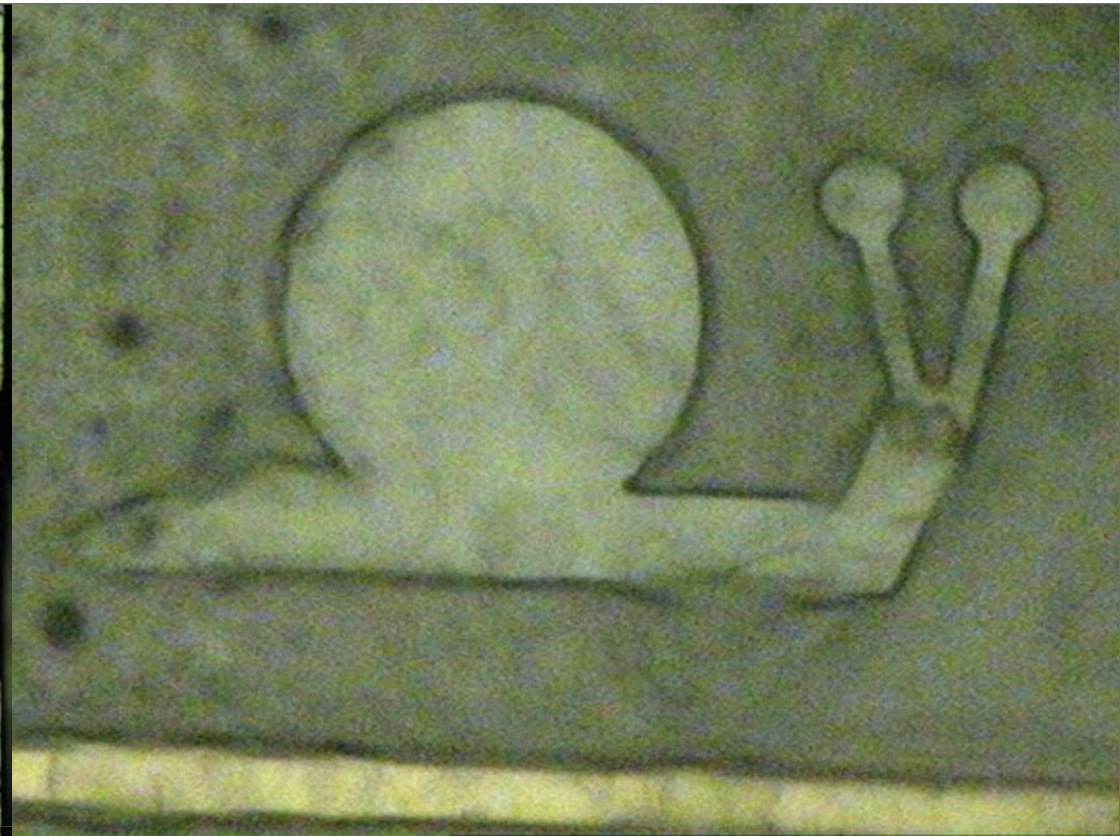
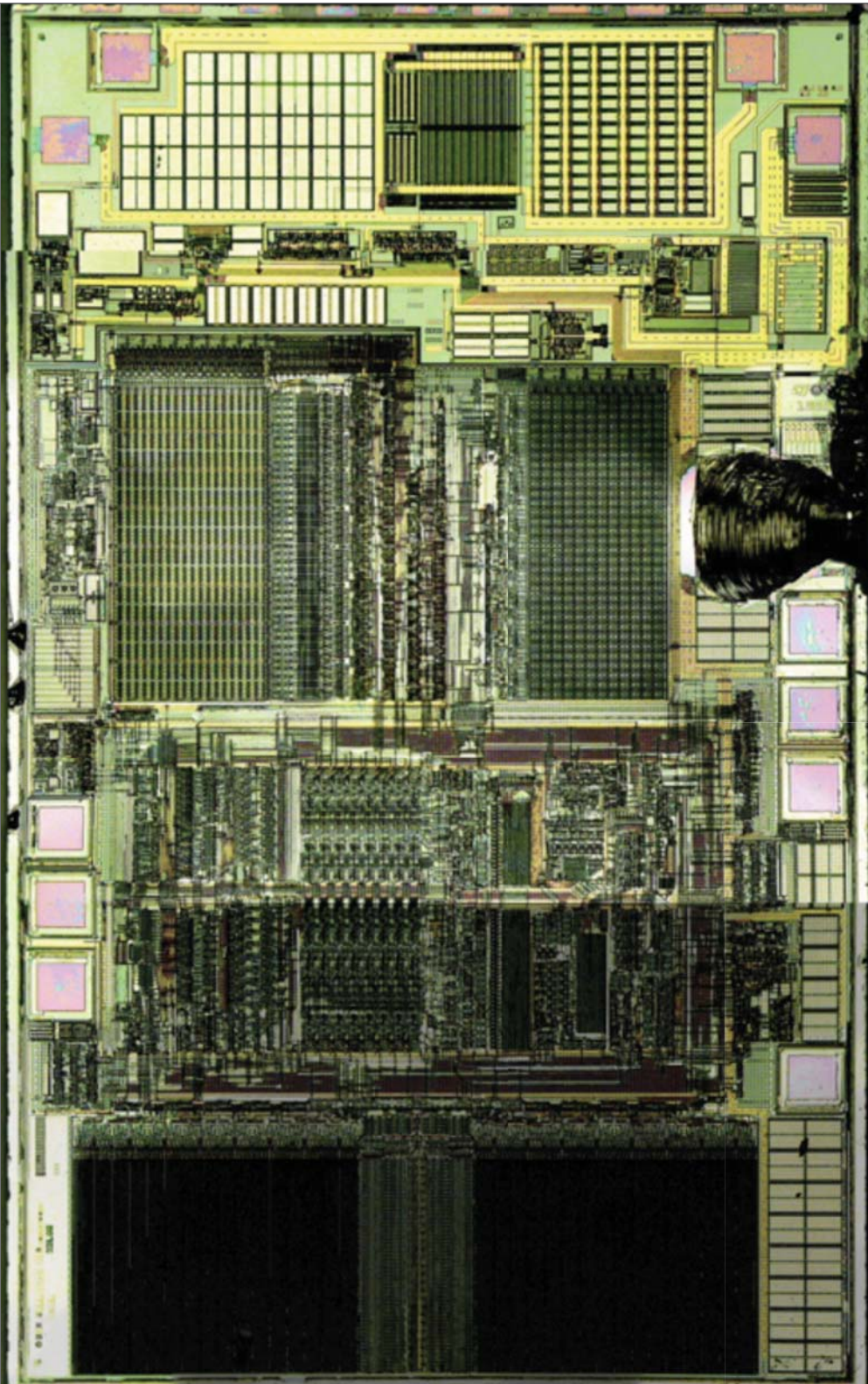


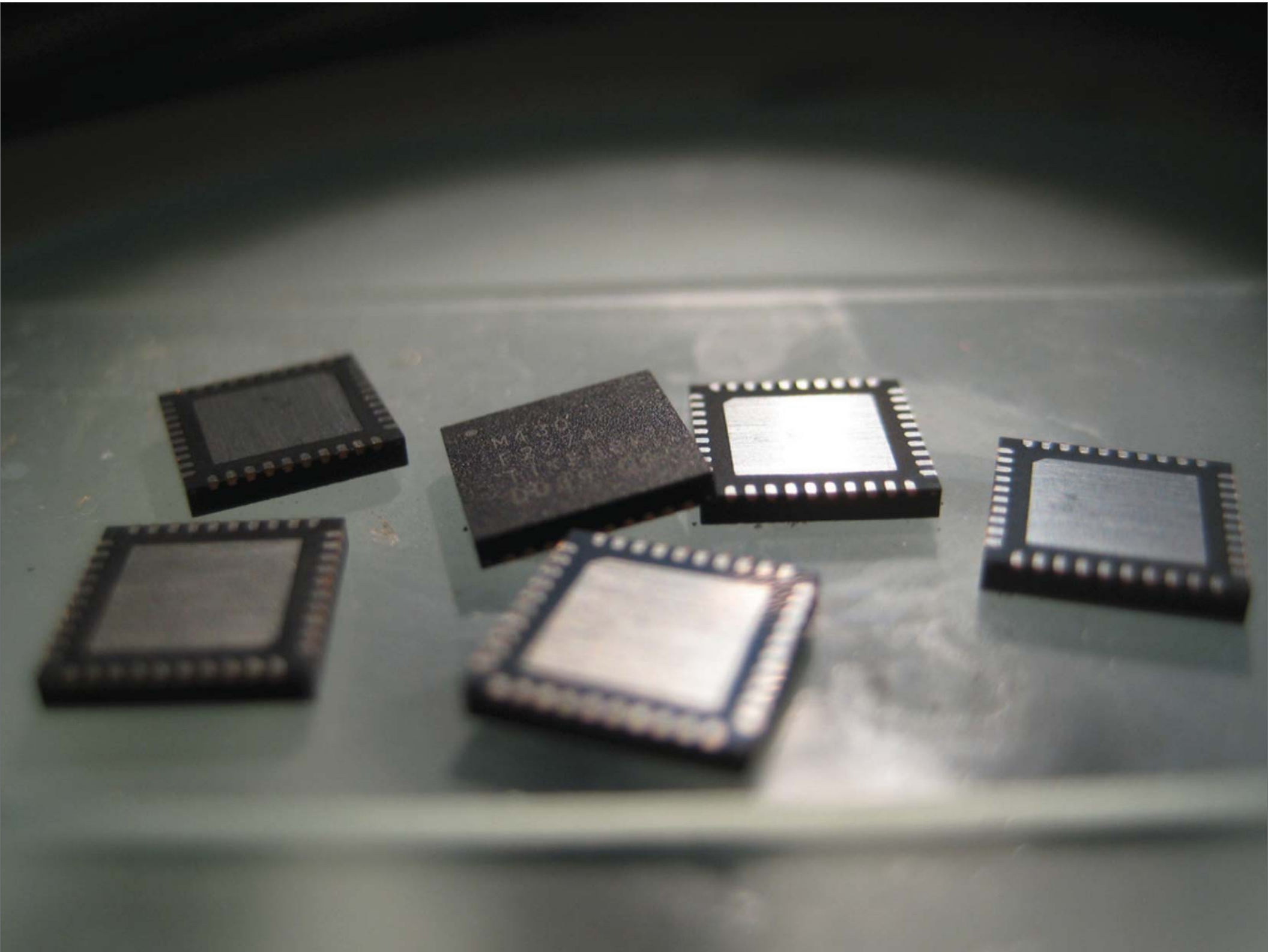
0 624 135 283 C

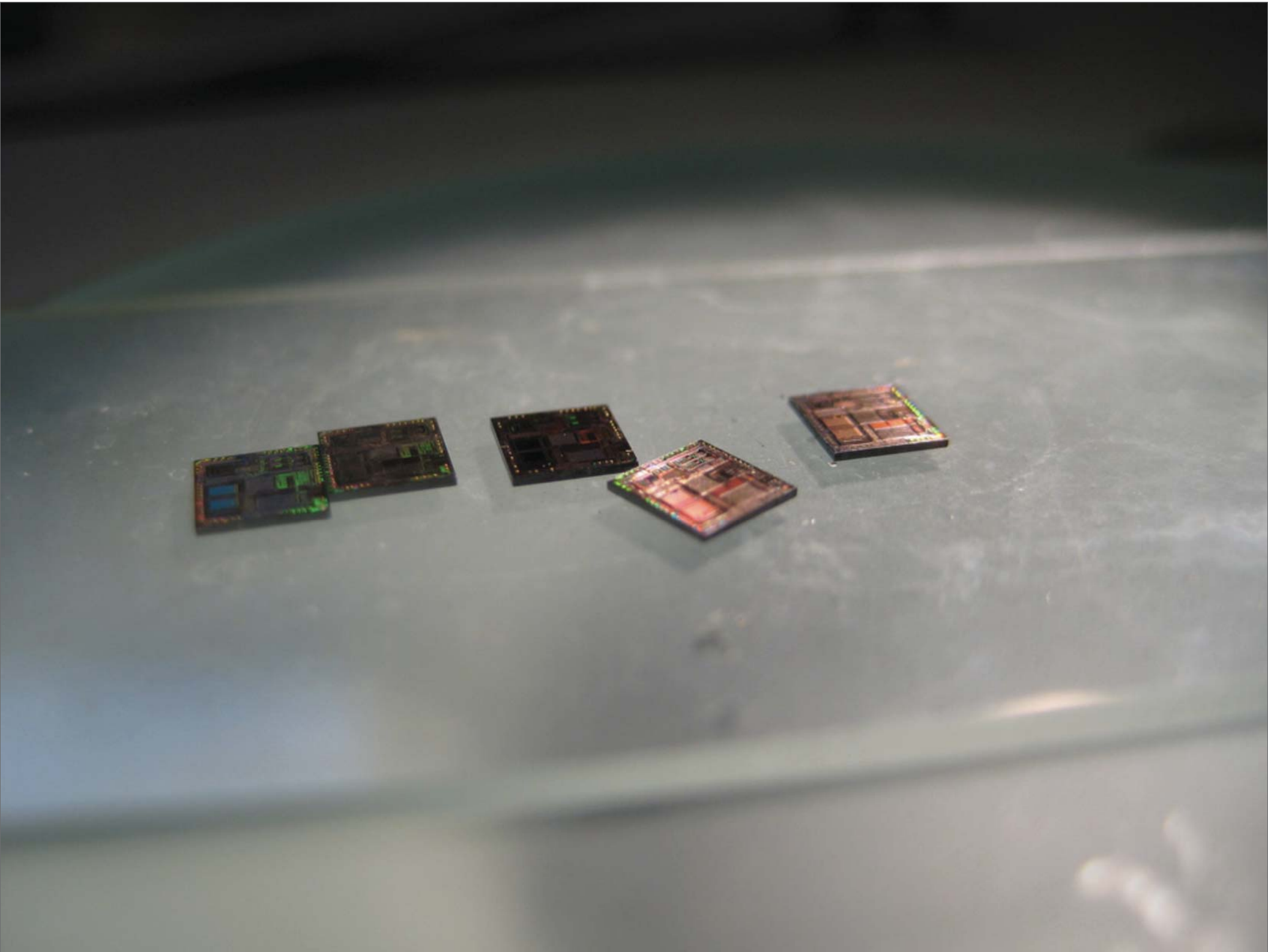


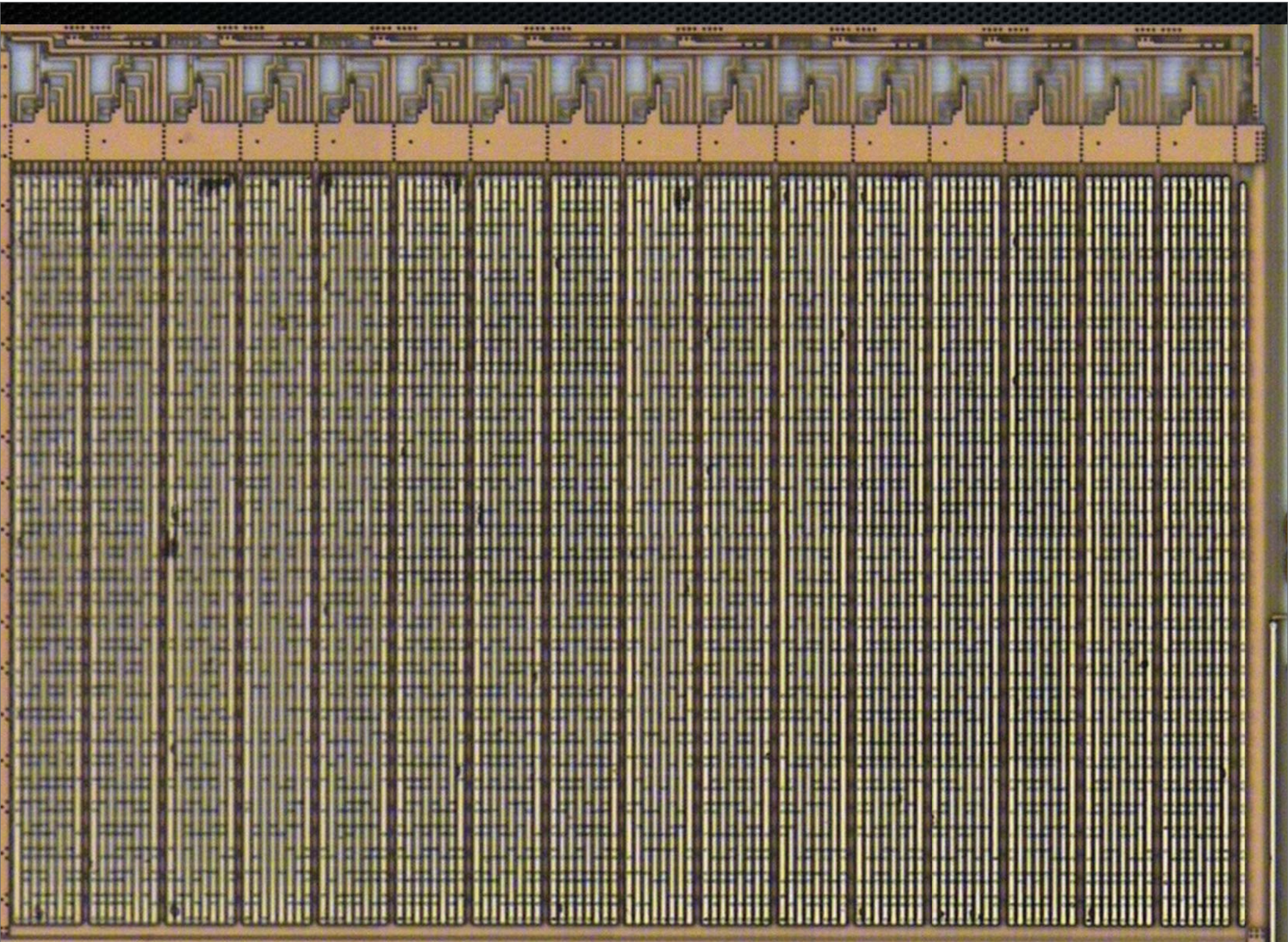


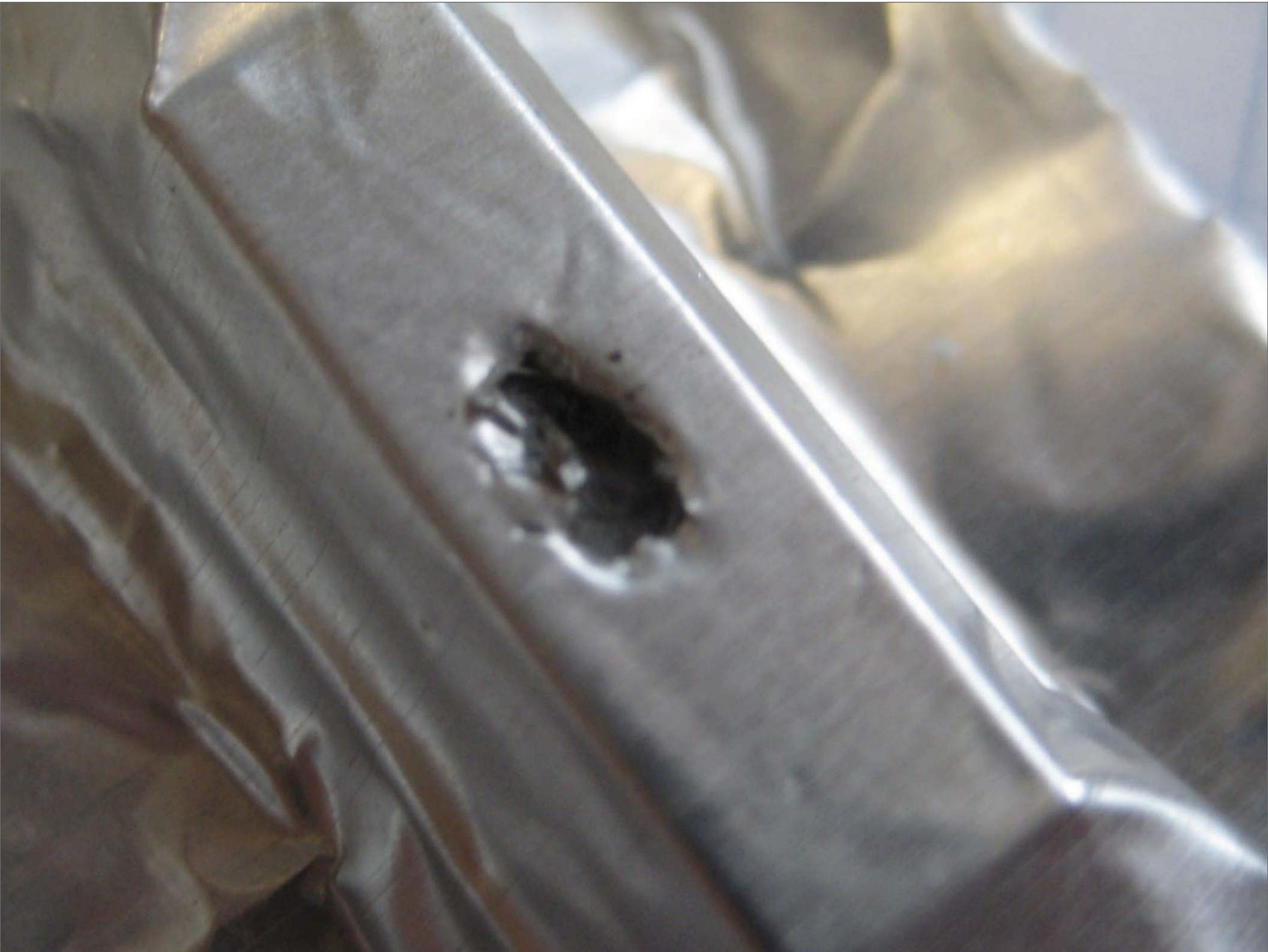
Friday, August 13, 2010



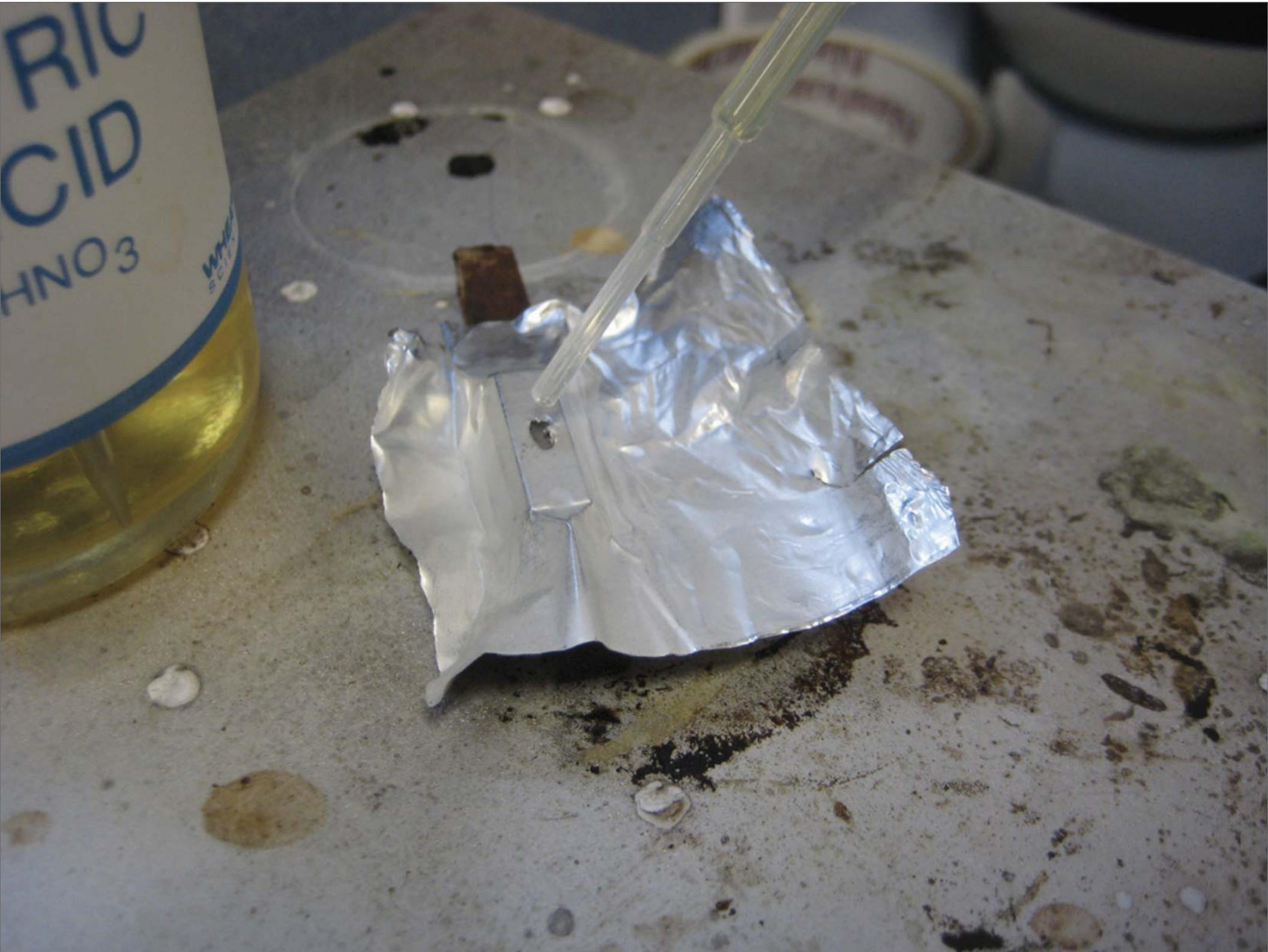








Friday, August 13, 2010

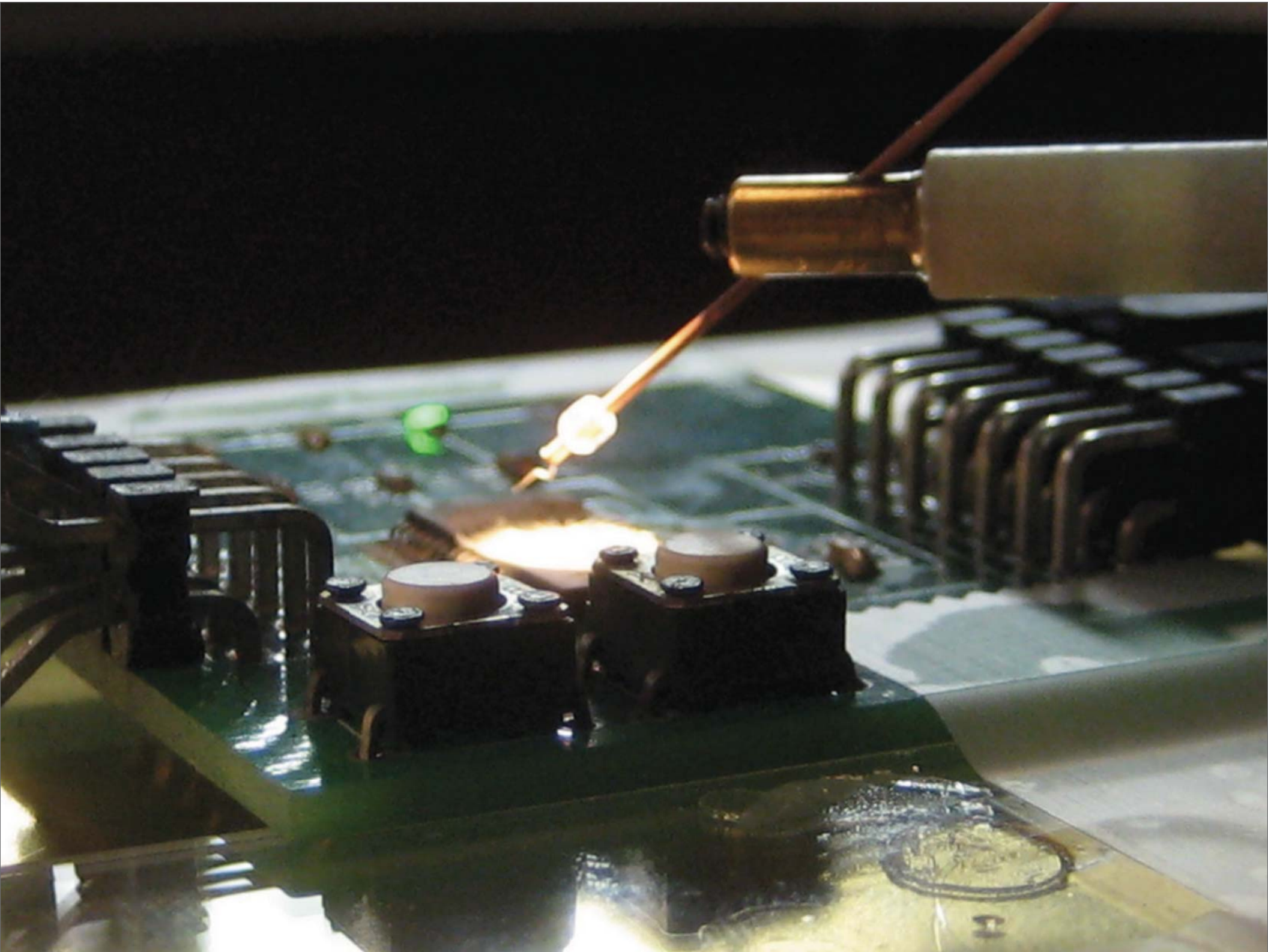


Friday, August 13, 2010

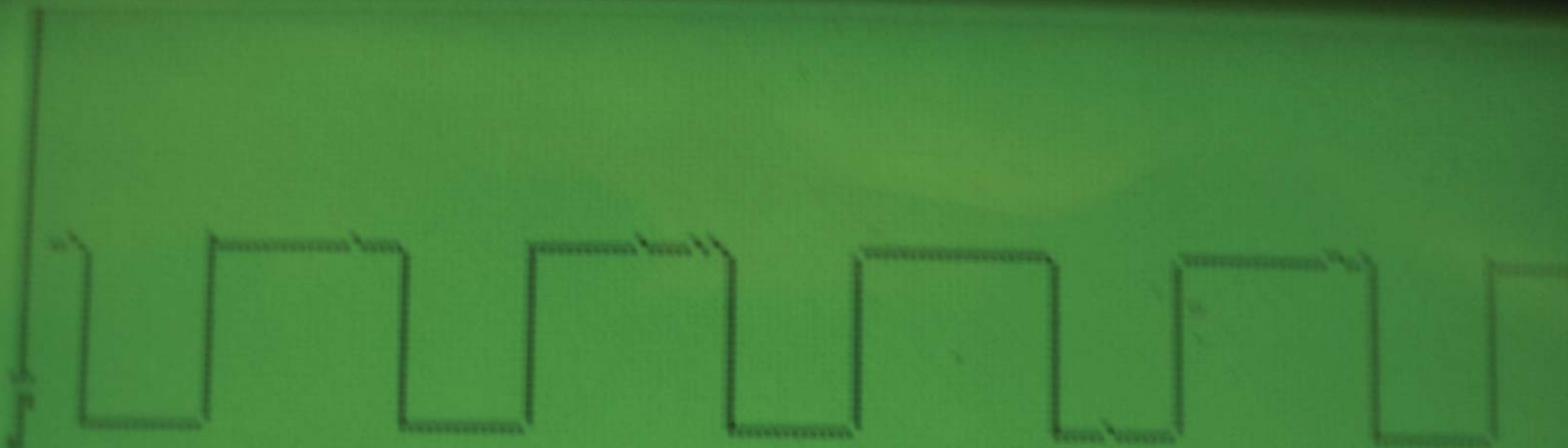


Friday, August 13, 2010





Friday, August 13, 2010



0.1s/div 0.25V/div 0.265